System	DDT Factor	@ 900 For Boot	1.4
16K	E080	00 29 33 00 00 3E	
46K	6880	00 A1 33 00 00 B6	
48K	6080	00 A9 33 00 00 BE	

```
                ;
                ;       CP/M v1.4 cbios for TEI FDC-2 Controller board
                ;
                ;       CUTER ROM at C000
                ;       Special disk rom at F000
                ;
                ;
002E =          msize   equ     46              ;memory in k
F000 =          dskrom  equ     0f000h
B600            bios    org     (msize*1024)-200h
A906 =          bdos    equ     bios-0d00h+6    ;bdos entry point
A100 =          ccp     equ     bios-1500h      ;command processor entry
002A =          nsects  equ     (bios-ccp)/128  ;length to warm boot
0003 =          iobyte  equ     3               ;special port byte
C019 =          sout    equ     0c019h          ;CUTER output
C01F =          sinp    equ     0c01fh          ;CUTER input
0014 =          comport equ     14h             ;comprint printer
0004 =          serport equ     4h              ;serial
                ;
                ;       standard bios transfer vector
                ;
B600 C333B6             jmp     boot            ;initial boot
B603 C358B6     wboot   jmp     warm            ;warm boot
B606 C363B7             jmp     const           ;console status
B609 C378B7             jmp     conin           ;console read
B60C C38EB7             jmp     conout          ;console write
B60F C3A9B7             jmp     list            ;list output
B612 C3CAB7             jmp     punch           ;punch output
B615 C3BEB7             jmp     reader          ;reader input
B618 C3E1B6             jmp     home            ;restore disk
B61B C3CAB6             jmp     select          ;select disk
B61E C3CFB6             jmp     settrk          ;set track number
B621 C3D4B6             jmp     setsec          ;set sector number
B624 C3D9B6             jmp     setdma          ;set buffer address
B627 C333B7             jmp     read            ;read disk
B62A C34BB7             jmp     write           ;write disk
B62D C32DB6             jmp     $
B630 C330B6             jmp     $
                ;
                ;       disk rom transfer vector
                ;
F003 =          dskboot equ     dskrom+3        ;general boot
F006 =          dskinit equ     dskrom+6        ;init controller
F009 =          dsksel  equ     dskrom+9        ;pysical select
F00C =          dskrest equ     dskrom+12       ;restore to track 0
F00F =          dskseek equ     dskrom+15       ;physical seek
F012 =          dskrd   equ     dskrom+18       ;read sector
F015 =          dskwr   equ     dskrom+21       ;write sector
F018 =          dskrt   equ     dskrom+24       ;read track
F01B =          dskwt   equ     dskrom+27       ;write track (format)
F01E =          dskrtrk equ     dskrom+30       ;read track #
F021 =          dskwtrk equ     dskrom+33       ;set track #
                ;
```

```
                ;       cold boot initialization
                ;
B633 =          boot    equ     $
B633 AF                 xra     a               ;init line printer
B634 D314               out     comport         ;init pia
B636 3D                 dcr     a
B637 D315               out     comport+1
B639 3E3E               mvi     a,3eh
B63B D314               out     comport
B63D DB15               in      comport+1       ;clear status
B63F 3EFF               mvi     a,0ffh          ;send a null
B641 D315               out     comport+1
B643 3E36               mvi     a,36h           ;toggle strobe bit
B645 D314               out     comport
B647 3E3E               mvi     a,3eh
B649 D314               out     comport
B64B 3E81               mvi     a,10000001b     ;initialize iobyte
B64D 320300             sta     iobyte
B650 3E00               mvi     a,0             ;clear char buffers
B652 3277B7             sta     char
B655 C39FB6             jmp     gocpm           ;static initialization
                ;
                ;       warm boot - read bdos and ccp
                ;
B658 =          warm    equ     $
B658 310001             lxi     sp,100h         ;reset stack
B65B CD06F0             call    dskinit         ;reset controller
B65E 0E00               mvi     c,0             ;drive 0 for boot
B660 CDCAB6             call    select
B663 CDE1B6             call    home
B666 C258B6             jnz     warm            ;no go...
B669 062A               mvi     b,nsects        ;sectors to read
B66B 0E00               mvi     c,0             ;current track
B66D 1602               mvi     d,2             ;current sector
B66F 2100A1             lxi     h,ccp           ;load address
B672 =          warm1   equ     $
B672 C5                 push    b               ;save registers
B673 D5                 push    d
B674 E5                 push    h
B675 4A                 mov     c,d             ;get sector
B676 CDD4B6             call    setsec
B679 C1                 pop     b               ;get dma address
B67A C5                 push    b               ;save for later
B67B CDD9B6             call    setdma          ;set read address
B67E CD33B7             call    read            ;read it
B681 C258B6             jnz     warm            ;error - try again
B684 E1                 pop     h               ;restore dma address
B685 118000             lxi     d,128           ;sector length
B688 19                 dad     d               ;new dma address in hl
B689 D1                 pop     d               ;get sector
B68A C1                 pop     b               ;restore counter, track
B68B 05                 dcr     b               ;1 less sector to go
B68C CA9FB6             jz      gocpm           ;all set
B68F 14                 inr     d               ;next sector
B690 7A                 mov     a,d
```

```
B691 FE1B               cpi     27              ;next track?
B693 DA72B6             jc      warm1           ;nope
B696 1601               mvi     d,1             ;reset sector to 1
B698 0C                 inr     c               ;increment track
B699 CDCFB6             call    settrk          ;set track number
B69C C372B6             jmp     warm1           ;and continue
                ;
                ;       transfer control to CP/M
                ;
B69F =          gocpm   equ     $
B69F 018000             lxi     b,80h           ;default buffer address
B6A2 CDD9B6             call    setdma
B6A5 3EC3               mvi     a,0c3h          ;set up jumps
B6A7 320000             sta     0
B6AA 320500             sta     5
B6AD 2103B6             lxi     h,wboot
B6B0 220100             shld    1
B6B3 2106A9             lxi     h,bdos
B6B6 220600             shld    6
B6B9 3EC3               mvi     a,0c3h          ;loop at rst 7
B6BB 323800             sta     70q
B6BE 213800             lxi     h,70q
B6C1 223900             shld    71q
B6C4 3E00               mvi     a,0             ;current drive to ccp
B6C6 4F                 mov     c,a
B6C7 C300A1             jmp     ccp             ;away we go.....
                ;
                ;       set desired drive
                ;
B6CA =          select  equ     $
B6CA 79                 mov     a,c
B6CB 32E1B7             sta     drvnxt
B6CE C9                 ret
                ;
                ;       set desired track
                ;
B6CF =          settrk  equ     $
B6CF 79                 mov     a,c
B6D0 32E7B7             sta     track
B6D3 C9                 ret
                ;
                ;       set desired sector
                ;
B6D4 =          setsec  equ     $
B6D4 79                 mov     a,c
B6D5 32E8B7             sta     sector
B6D8 C9                 ret
                ;
                ;       set desired buffer address
                ;
B6D9 =          setdma  equ     $
B6D9 E5                 push    h               ;save registers
B6DA 60                 mov     h,b
B6DB 69                 mov     l,c
B6DC 22E9B7             shld    dma
```

```
B6DF E1                 pop     h               ;restore
B6E0 C9                 ret
                ;
                ;       send a disk (physically) home
                ;
B6E1 =          home    equ     $
B6E1 0E00               mvi     c,0             ;set track to 0
B6E3 CDCFB6             call    settrk
B6E6 CDEDB6             call    seldsk          ;select the disk
B6E9 CD29B7             call    seek            ;do the seek
B6EC C9                 ret
                ;
                ;       select disk for transfer
                ;
B6ED =          seldsk  equ     $
B6ED C5                 push    b               ;save registers
B6EE 3AE0B7             lda     drvcur          ;what was last drive?
B6F1 4F                 mov     c,a
B6F2 3AE1B7             lda     drvnxt          ;what is desired drive?
B6F5 B9                 cmp     c               ;save drive?
B6F6 CA21B7             jz      seldsk2         ;yes
B6F9 E5                 push    h               ;save register
B6FA 21E2B7             lxi     h,drvtrk        ;point to track table
B6FD 0600               mvi     b,0
B6FF 09                 dad     b               ;index by current drive
B700 CD1EF0             call    dskrtrk         ;get head position
B703 77                 mov     m,a             ;save current position
B704 3AE1B7             lda     drvnxt          ;get desired drive
B707 4F                 mov     c,a
B708 CD09F0             call    dsksel          ;select it for real
B70B 21E2B7             lxi     h,drvtrk        ;point to track table
B70E 09                 dad     b               ;index by drive
B70F 7E                 mov     a,m             ;retrieve its position
B710 4E                 mov     c,m
B711 FEFF               cpi     0ffh            ;is it known??
B713 C21CB7             jnz     seldsk1         ;yes it is
B716 CD0CF0             call    dskrest         ;no - calibrate
B719 0E00               mvi     c,0             ;set track to 0
B71B 71                 mov     m,c
B71C =          seldsk1 equ     $
B71C 79                 mov     a,c             ;get position
B71D CD21F0             call    dskwtrk         ;tell the controller
B720 E1                 pop     h               ;restore register
B721 =          seldsk2 equ     $
B721 3AE1B7             lda     drvnxt          ;set current drive
B724 32E0B7             sta     drvcur
B727 C1                 pop     b               ;restore registers
B728 C9                 ret
                ;
                ;       do a seek
                ;
B729 =          seek    equ     $
B729 C5                 push    b               ;save register
B72A 3AE7B7             lda     track           ;get desired track
B72D 4F                 mov     c,a
```

```
B72E CD0FF0             call    dskseek         ;do the seek
B731 C1                 pop     b               ;restore register
B732 C9                 ret
                ;
                ;       do a read
                ;
B733 =          read    equ     $
B733 E5                 push    h               ;save registers
B734 C5                 push    b
B735 CDEDB6             call    seldsk          ;select disk
B738 CD29B7             call    seek            ;position head, load
B73B C248B7             jnz     read1           ;error
B73E 2AE9B7             lhld    dma             ;get buffer address
B741 3AE8B7             lda     sector          ;get desired sector
B744 4F                 mov     c,a
B745 CD12F0             call    dskrd           ;do the read
B748 =          read1   equ     $
B748 C1                 pop     b               ;restore registers
B749 E1                 pop     h
B74A C9                 ret
                ;
                ;       do a write
                ;
B74B =          write   equ     $
B74B E5                 push    h               ;save registers
B74C C5                 push    b
B74D CDEDB6             call    seldsk          ;select drive
B750 CD29B7             call    seek            ;position head, load
B753 C260B7             jnz     write1          ;error
B756 2AE9B7             lhld    dma             ;get buffer address
B759 3AE8B7             lda     sector          ;get desired sector
B75C 4F                 mov     c,a
B75D CD15F0             call    dskwr           ;write it out
B760 =          write1  equ     $
B760 C1                 pop     b               ;restore registers
B761 E1                 pop     h
B762 C9                 ret
                ;
                ;       keyboard status
                ;
                const:
B763 3A77B7             lda     char            ;see if char present
B766 B7                 ora     a
B767 3E00               mvi     a,0             ;set up for cma
B769 C275B7             jnz     kbdst1          ;already have a char
B76C CD1FC0             call    sinp            ;call status/get
B76F 3277B7             sta     char            ;save char if any
B772 3E00               mvi     a,0             ;clear status
B774 C8                 rz
                kbdst1:
B775 2F                 cma                     ;no char available
B776 C9                 ret
B777 00         char:   db      0               ;character buffer
                ;
                ;       keyboard input
```

```
                        ;
                        conin:
B778 3A77B7                     lda     char            ;get char
B77B B7                         ora     a               ;test if present
B77C C285B7                     jnz     kbdin1          ;have char already
B77F CD63B7                     call    const           ;test for char
B782 C378B7                     jmp     conin           ;and try again
                        kbdin1:
B785 4F                         mov     c,a             ;save char
B786 AF                         xra     a               ;clear buffer
B787 3277B7                     sta     char
B78A 79                         mov     a,c             ;restore char
B78B E67F                       ani     7fh             ;strip parity
B78D C9                         ret
                        ;
                        ;       video output
                        ;
                        conout:
B78E C5                         push    b               ;save registers
B78F 41                         mov     b,c             ;current char in b
B790 3AA3B7                     lda     lastchar        ;get last char
B793 CDA4B7                     call    checkcr         ;ignore duplicate cr
B796 CAA0B7                     jz      noprint
B799 79                         mov     a,c             ;get char in a
B79A 32A3B7                     sta     lastchar        ;save as last
B79D CD19C0                     call    sout            ;display char
                        noprint:
B7A0 C1                         pop     b               ;restore registers
B7A1 79                         mov     a,c             ;set a to char
B7A2 C9                         ret                     ;and return
B7A3 00                 lastchar db     0               ;last char displayed
                        ;
                        ;       routine to check for two cr in a row
                        ;
                        checkcr:
B7A4 FE0D                       cpi     0dh             ;was last char a cr
B7A6 C0                         rnz                     ;no, print this one
B7A7 B8                         cmp     b               ;is this one a cr too?
B7A8 C9                         ret                     ;return with status
                        ;
                        ;       printer output
                        ;
                        list:
B7A9 DB14                       in      comport         ;get status
B7AB 07                         rlc                     ;test status
B7AC D2A9B7                     jnc     list            ;wait
B7AF DB15                       in      comport+1       ;clear status
B7B1 79                         mov     a,c             ;get char
B7B2 2F                         cma                     ;invert for comprint
B7B3 D315                       out     comport+1
B7B5 3E36                       mvi     a,36h           ;toggle strobe bit
B7B7 D314                       out     comport
B7B9 3E3E                       mvi     a,3eh
B7BB D314                       out     comport
B7BD C9                         ret
```

```
                        ;
                        ;       serial input
                        ;
                        reader:
B7BE DB04               in      serport         ;get status
B7C0 E601               ani     1               ;test status
B7C2 C2BEB7             jnz     reader          ;wait
B7C5 DB05               in      serport+1       ;get char
B7C7 E67F               ani     7fh             ;clear parity
B7C9 C9                 ret
                        ;
                        ;       serial output
                        ;
                        punch:
B7CA DB04               in      serport         ;get status
B7CC E680               ani     80h             ;test status
B7CE C2CAB7             jnz     punch           ;wait
B7D1 79                 mov     a,c             ;get char
B7D2 D304               out     serport         ;send it
B7D4 C9                 ret
                        ;
                        ;       scratch area
                        ;
B7E0                    scratch org     msize*1024-32   ;top of memory
B7E0 05                 drvcur  db      5               ;max drive + 1
B7E1 00                 drvnxt  db      0
B7E2 FFFFFFFFFF         drvtrk  db      0ffh,0ffh,0ffh,0ffh,0ffh  ;track table
B7E7 00                 track   db      0               ;desired track
B7E8 00                 sector  db      0               ;desired sector
B7E9 8000               dma     dw      80h             ;buffer address
B7EB                    end
```